

IDENTIFICATION AND DETECTION OF ANOMALIES THROUGH SSME DATA ANALYSIS

Lisa Pereira

Moonis Ali

Center For Advanced Space Propulsion
The University of Tennessee Space Institute
Tullahoma, TN 37388

Abstract

The goal of the ongoing research described in this paper is to analyze real-time ground test data in order to identify patterns associated with the anomalous engine behavior, and on the basis of this analysis to develop an expert system which detects anomalous engine behavior in the early stages of fault development. A prototype of the expert system has been developed and tested on the high frequency data of two SSME tests, namely Test #901-0516 and Test #904-044. The comparison of our results with the post-test analyses indicates that the expert system detected the presence of the anomalies in a significantly early stage of fault development.

1. Introduction

A complex physical system, such as the Space Shuttle Main Engine (SSME), is subject to component failures at any time during its testing or operation. When a fault does occur, it is most likely that its effects will lead to damages in its components, possibly including partial or total destruction of the SSME. While testing the SSME, engineers are not able to see the onset of a fault, so when the fault fully develops, it could cause either early engine shutdown or serious damage to the SSME. Worse scenarios could happen if a slowly occurring fault goes undetected throughout all of the SSME's ground and flight tests, and once the engine is inserted into the shuttle for an actual flight, the fault fully develops, causing catastrophic damage to the shuttle and the crew.

When a fault does occur, extensive post-test data analysis is performed by examining a set of sensor plots from different time slices in order to determine a fault's behavioral characteristics, such as the time the sensor data started to indicate the faulty behavior and how the sensor data showed the fault's trend throughout its development. Many times some of the sensors start indicating abnormal SSME behavior

considerably before the fault is noticeable by the engineers and before the parameters set for the redline criteria are met, especially if it is a slowly occurring fault. If the engineers or flight crew have an early indication of the developing anomalous engine behavior, preventative measures can be taken in order to avoid or minimize the consequences of the fault.

Automating the sensor data analysis process performed by engineers will result in an online detection system that can discover faults in a physical system during their early development stages by noticing the behavioral changes in all of the sensors' data. Such a system can be a useful tool in aiding engineers during a test, since it would warn them about the onset of anomalies occurring in the SSME during the test as opposed to afterward when a fault might cause early engine shutdown and damage to the SSME.

When analyzing the sensor data, engineers integrate the results from several sensors in order to come up with a more substantial hypothesis or conclusion about what has occurred with the SSME. By integrating the results from all of the generated sensor hypotheses, a detection expert system can provide a better and more precise indication of the health of a rocket engine, thereby diminishing the possibility of false alarms located generated by noise found in the data or by sensors located away from the monitored component.

Many diagnostic expert systems have been developed for rocket and jet engine domains [1-8]. The following is a review of the different approaches taken by researchers in solving fault detection and diagnostic problems.

The system described in [2] uses explicit knowledge representation and explicit reasoning to detect and diagnose faults in a jet engine. The detection of faults is based on stored explicit knowledge incorporated into each node in the domain dependent diagnostic tree. If a certain symptom matches a node in the tree, then reasoning rules, which are specific to a given node, are applied in order to guide the traversal through the fault diagnosis tree.

This research was supported by the System Dynamics Laboratory of NASA-MSFC Grant No. NAG8-121.

Rule-based diagnostic systems were used by the systems in [5,8]. In the system described in [5], the detection of faults is done by looking for abnormalities based on analytical redundancy contained in the Kalman Filter. The diagnostic process involves two approaches, one in which the origin of a failure is determined by the elements most likely to have caused the problem, and the other in which the mathematical modeling corresponds to aircraft configuration changes due to the origins of a fault. Rules are used for transforming the scheduled tasks and failure accommodation into a search problem, which schedules and selects the actions taken by the system. In the second system [8] rules are used to actually perform the diagnosis of a fault. It has two approaches in diagnosing faults. The first approach involves fault modeling based on the operation of the engine; it identifies the problems and determines the fault following trouble-shooting procedures. In the second approach, a qualitative model is used to generate fault hypothesis; the system then chooses one of the hypothesis and looks into the physical layout in order to infer which problems could occur. If the problems do not match the symptoms determined by the system, then that given hypothesis is eliminated.

The system described in [7] uses qualitative and temporal data to perform diagnostics in rocket engine data. Comparisons between the input and previously seen data are executed by the reasoning processor, and similarities found are used to diagnose abnormalities found in the sensor data. In this way, the system can detect faults that have not yet exceeded the safety parameters set for the rocket engine.

Another fault diagnostics approach implemented in [1] is a diagnostic system that combines two parallel approaches when detecting and diagnosing anomalous behavior in propulsion systems. In the first approach, the system learns and identifies sensor data behavior patterns, and generates hypothesis based on the behavior of the system. At the same time, the subsystem involved with the second approach processes the sensor data and reasons with the processed data, the design and functional knowledge of the propulsion system, and the knowledge of the principles of physics and mechanics of the propulsion system in order to generate a fault hypothesis. Results from both approaches are then integrated to form a final hypothesis about the propulsion system.

Neural networks have also been used as a method for detecting and diagnosing faults in rocket and jet engines [3,4,6]. They analyze temporal rocket or jet engine data represented in the form of sensor data curves. In [3] faults are detected and diagnosed by matching an incoming curve with known stored patterns with which the neural network has been trained, whereas in [4,6] detection and diagnosis is performed

by looking at the activation values of the middle layer nodes.

The goal of the ongoing research described in this paper is to analyze real-time ground test data, to identify patterns associated with the anomalous engine behavior, and on the basis of this analysis to develop an "Identification and Detection Expert System (IDES) which detects anomalous engine behavior in the early stages of fault development significantly earlier than the indication provided by the redline detection mechanism. A prototype of IDES has been developed and tested on the high frequency data of the two tests where anomalous behavior of a High Pressure Oxidizer Turbo-Pump (HPOTP) was the cause of the fault. IDES's detection approach is based on the methodology applied by rotordynamics experts when they analyze the post-test high frequency sensor data. The system is designed to look for any kind of anomalies present in the monitored information found in the sensor data. The system also integrates each sensor's information into a single hypothesis about what the sensor sees as the behavior of the HPOTP, and then all of the generated sensor hypothesis are integrated in order to determine whether there is an actual fault occurring, or whether one of the sensors is just picking up feed through frequencies from other components.

In order to detect HPOTP faults more accurately and in their very early developing stages, the Isolator and Weld Strain Gage sensors were selected for monitoring. These sensors have been determined by the experts to be the best indicators of HPOTP faults because they are the closest sensors to the source of the problems we have analyzed in the two tests. The Isolator Strain Gages, when present in the HPOTP, are internal to the pump, while the Weld Strain Gages are located on the outside casing of the HPOTP.

It should be noted that the sensor data is not in a steady state from the beginning to the end because of scheduled events that have an impact on the data, causing sensor data to become transient for a while. In order to deal with transient state sensor data, the system utilizes knowledge about the scheduled thrust level changes in order to determine if the sensor data should be sampled and processed. If the system determines that the current time slice falls within the non-transient monitoring time period, it allows for the sensor data to be sampled and analyzed; otherwise, the system ignores the sensor data for that time slice and waits for the next possible processing time slice. The system also keeps information on the scheduled vehicle commands and informs the user whenever a scheduled event occurs during the safe monitoring period, so that if an anomaly is detected, the user is made aware that it could have been caused by the scheduled event.

The system also monitors and identifies intermittent frequencies found in a sensor's window of data. This allows for the system to warn engineers about frequencies that are significantly appearing in the sensor data, possibly indicating faults in other components of SSME. While some of the intermittent frequencies are known to engineers, others are of an unknown nature, giving engineers more information about HPOTP's behavior. The system also tracks and informs engineers about intermittent frequencies which it has detected in the past for each of the monitored sensors. In this way engineers can see which intermittent frequencies are always appearing in a sensor's data.

II. Expert System Architecture

The architecture of the Identification and Detection Expert System (IDES) is shown in Figure 1. It is comprised of four modules. These modules are: the Monitor, the Frequency Extractor, the Data Analyzer/Fault Detector, and the Sensor Integrator. The Profile of Scheduled Events provides information to the Monitor in order to allow IDES to differentiate anomalies from scheduled events. A user interface is also integrated with IDES in order to facilitate the interaction between the user and the system.

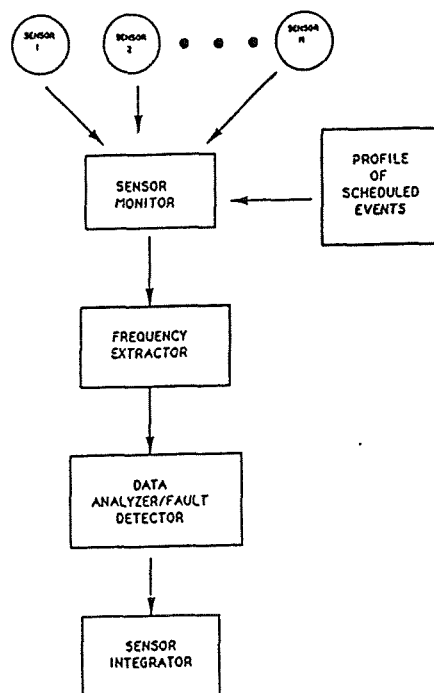


Figure 1. The System's Architecture.

The Monitor

The Monitor is responsible for sampling sensors from sequential windows of 0.4 seconds duration from the non-transient portion of data. The non-transiency of data is determined by utilizing the information provided by the Profile of Scheduled Events. A fast fourier transform of the data of each window is considered for further analysis. The Monitor is also responsible for checking with the Profile of Scheduled Events in order to avoid mistaking a scheduled event for an anomaly. The interface also informs users about the temporal events when a thrust level change command or any other scheduled vehicle command is executed.

The Frequency Detector

This module is responsible for analyzing each window of data and detecting the presence of frequencies of unusually high amplitude. First, attempts are made to identify these frequencies as some of the known frequencies. The frequencies which cannot be identified as known frequencies are treated as intermittent frequencies. Through the user interface, experts/operators are informed about the presence of these abnormal activities in the data.

The Data Analyzer/Fault Detector

The purpose of this module is to perform a comparative analysis of the newly extracted values of the monitored sensor information in order to determine if any of them are detecting anomalous HPOTP behavior. Given each of the extracted values and their respective expected normal values, the Data Analyzer/Fault Detector (DA/FD) calculates an abnormality score for each value, representing the degree by which it is detecting anomalous behavior. Based on the calculated abnormality scores, DA/FD decides whether an anomaly exists in any monitored information of a given sensor. When processing normal data, this module is also made responsible for learning what the expected normal values for the sensor information are.

The Sensor Data Integrator

This module has two components that are responsible for performing sensor data integration. The first component integrates all the available information on a sensor's monitored frequencies and generates a consistency score, which states how consistently any of the monitored sensor information has been detecting anomalies. After each sensor has been processed, the second component then integrates all of the generated sensor consistency hypotheses into a single overall hypothesis about whether the HPOTP's current status is normal or if any of the sensors are showing anomalous behavior.

The User Interface

In order to organize the display of IDES's output and to simplify a user's interaction with IDES, a user interface was integrated with the overall system. The layout for the interface consists of four sensor analysis output display windows and a set of system commands that allow the user to interact with IDES, as shown in Figure 2.

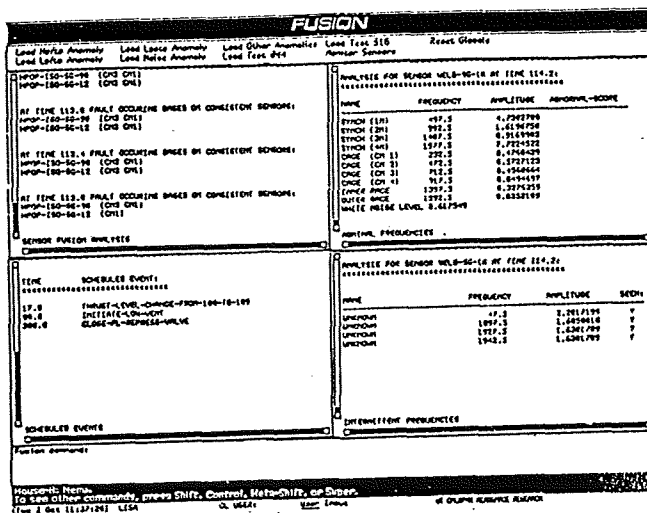


Figure 2. The User Interface.

Each of the windows shows a specific set of information that has either been extracted from the sensor data, supplied by the Profile of Scheduled Events, or deduced during IDES's reasoning process. The Nominal Frequency Window (NFW) and the Intermittent Frequency Window (IFW) display the frequency names, the frequency values, and the frequency amplitudes for all of the frequencies displayed in the respective windows. In the NFW, if a frequency is detecting an anomaly, its abnormality score is also output to the screen. This window also displays the white noise value for each of the sensors. In IFW, if any of the currently extracted frequencies have been seen before by the respective sensor, then it informs the user of that by printing a Y under the Seen column. Any time that a vehicle command is scheduled during IDES's current monitoring time slice, it is displayed in

the Scheduled Events Window, along with the previously and next scheduled events. In the Sensor Fusion Analysis Window, the final hypothesis generated by IDES about the HPOTP is displayed for each monitored time slice; the sensors that contributed to the final hypothesis have their names indicated under the hypothesis.

The set of system commands displayed on the top of the interface's screen allows a user to interact with IDES without knowing any of the required parameters and function calls needed to run IDES. These commands provide the user with the flexibility of loading different test cases to be run through IDES. They are activated by either clicking the mouse on the desired option, or by typing a desired command in the command line.

III. Scheduled Events and Monitoring Strategy

IDES maintains a profile of scheduled events. The information contained in this profile is used by the Monitor in order to determine SSME's transient time period between scheduled thrust level changes, the new thrust level, and any scheduled vehicle commands that may affect the amplitude values of the monitored frequencies. The Profile of Scheduled Events is composed of two parts, one which contains the available information on the thrust level events, and the other which contains the available information on all the vehicle commands scheduled for a given SSME test.

A change in thrust level causes a transient state in the sensor data and brings instability to the frequency and amplitude relationship. These instabilities may mislead the system to erroneous frequency analysis. This is especially true if the thrust level change is drastic, as when it changes from 104 to 65 %. In order to avoid erroneous results, IDES employs a scheduled event profile to skip the transient and unstable period of data and analysis data of non-transient and stable durations only.

At each thrust change, the Profile of Scheduled Thrust Level Change Events (PSTLCE) provides the IDES with the SSME's new thrust level, and the stable data monitoring start and end times for the given thrust level. With this information IDES determines the time duration during which it can sample the sensor data.

Although anomalous behavior found in the sensor data usually indicates the development of an anomaly, it could be caused as an after-effect of a vehicle command scheduled to occur at a given time. In order to account for anomalous sensor data behavior, the Profile of Scheduled Vehicle Events provides the user with the current scheduled event, the previously

scheduled event, and the next scheduled event. If anomalous behavior is detected during a time scheduled for a vehicle command, the user is informed that the event may have had an effect on the sensor data, especially if some of the sensors detect anomalies in their monitored frequencies and no fault hypothesis is generated for the next few seconds.

As explained earlier, the monitor is responsible for sampling sensors from sequential windows of 0.4 seconds duration. The frequency spectrum of a window of strain gauge 12 data is given in Figure 3 which illustrates amplitudes of all frequencies between 0 and 2000 Hz. Each frequency in this spectrum represents the mid point of the 5 Hz bin which is used in sampling the data by the monitor.

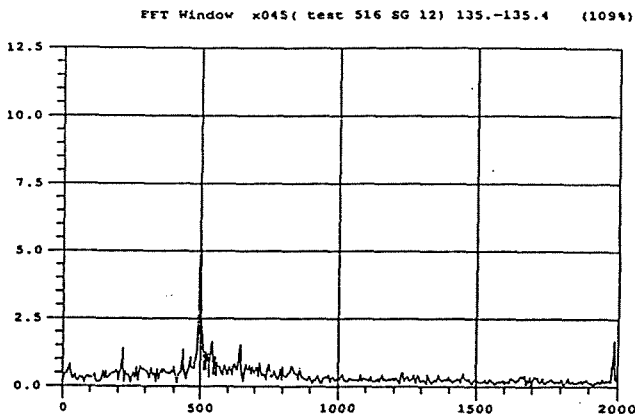


Figure 3. A Window of Sensor Data

IV. The Frequency Extractor (FE)

Human experts employ certain specific frequencies and their harmonics (termed as basic frequencies in this paper) in analyzing anomalous engine behavior. A list of such frequencies is given in Table 1. The presence of high amplitude at any arbitrary frequencies other than basic frequencies (termed as intermittent frequencies in this paper) may also indicate the presence of an anomaly. The Frequency Extractor, composed of two modules, scans each sensor data window and identifies and retrieves the needed frequency information for the analysis of HPOTP's health status. The two modules of FE are the Basic Frequency Extractor and the Intermittent Frequency Extractor, and each is described in the following subsections.

Design of the Basic Frequency Extractor

The Basic Frequency Extractor (BFE) design is

MONITORED SENSOR DATA INFORMATION	
1. FUNDAMENTAL SYNCHRONOUS FREQUENCY	(1N)
2. SECOND HARMONIC OF SYNCHRONOUS FREQUENCY	(2N)
3. THIRD HARMONIC OF SYNCHRONOUS FREQUENCY	(3N)
4. FOURTH HARMONIC OF SYNCHRONOUS FREQUENCY	(4N)
5. FUNDAMENTAL CAGE FREQUENCY	(1X)
6. SECOND HARMONIC OF CAGE FREQUENCY	(2X)
7. THIRD HARMONIC OF CAGE FREQUENCY	(3X)
8. FOURTH HARMONIC OF CAGE FREQUENCY	(4X)
9. INNER RACE FREQUENCY	(IR)
10. OUTER RACE FREQUENCY	(OR)
11. WHITE NOISE LEVEL	

Table 1. Extracted Sensor Data Information

based on the heuristics applied by the experts when they identify the synchronous and cage fundamental frequencies, their respective harmonics, and the inner and outer race frequencies. Each frequency has a certain range in the data window where it can be found, and it is identified as the highest amplitude peak within that range. If there is not an apparent peak found within the expected frequency range, the given frequency is said to be absent at that current data window, or not significantly appearing in the sensor data.

When analyzing sensor data at different thrust levels, one can notice a shift in the frequency locations as shown in Figure 4. BFE needed a way to determine the necessary frequency ranges so that it can search for the desired frequencies, regardless of SSME's current thrust level. Sensor data was not available for all the possible thrust levels, thus eliminating the possibility of containing a table with frequency ranges for all the thrust levels. Another possibility could have been to keep ranges wide enough so that the desired frequencies could be found no matter what the thrust level is, but this method could produce wrong frequency identifications due to overlapping frequency ranges, such as the fundamental synchronous frequency and the first cage harmonic ranges, and to aliasing of frequencies around a given frequency's range.

This frequency range problem was solved by first manually analyzing several windows of normal data at different thrust levels, extracting the fundamental synchronous frequency from each of the windows, and then fitting the frequency points through a second degree polynomial curve that determines the approximate location of the first synchronous frequency

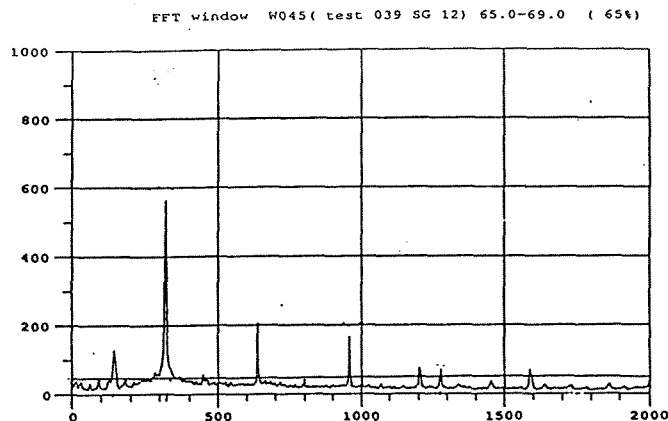
at any given thrust level. In this way, in analyzing a new test data, a shorter range of $\pm 30\text{Hz}$ is put around the approximate first synchronous frequency location (determined from the second degree polynomial) to insure that if the synchronous frequency has shifted, it can still be found within the expected range, especially since error is introduced by the sampling rate of the data and by the approximation of the interpolated function. Once the fundamental synchronous frequency has been identified for a given thrust level, a range of $\pm 10\text{Hz}$ is put around it, creating a shorter range by which BFE must search for the fundamental synchronous frequency in subsequent windows. This process was not repeated for the other frequencies since their ranges are determined by the correct identification of the fundamental synchronous frequency.

In order to determine the frequency ranges for the synchronous harmonics, BFE generates a range of $\pm 10\text{Hz}$ around the first synchronous frequency. It then multiplies this range by two, three, and four to get the first, second, and third harmonic ranges respectively.

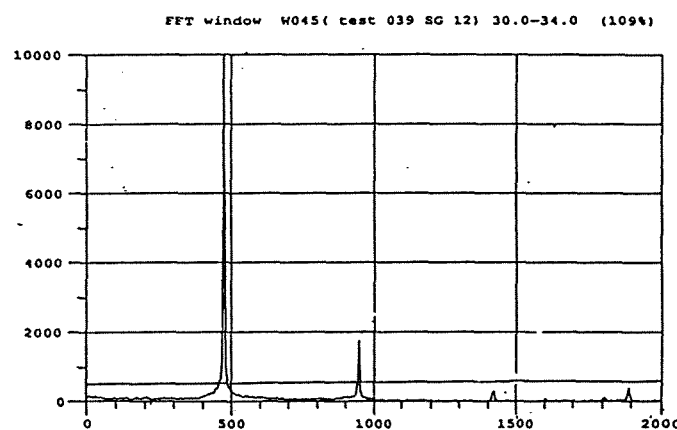
Experts in the field have determined that the fundamental cage frequency is always found within a range of 42-47 % of the fundamental synchronous frequency. The system computes this range and identifies the fundamental cage frequency as the highest peak in that range. Once the first cage frequency is identified, BFE generates a range of $\pm 10\text{Hz}$ around it, and multiplies the new range by two, three, and four in order to generate the first, second, and third cage harmonic ranges.

Once all of the harmonic ranges are found, BFE matches the synchronous harmonics to the fundamental synchronous frequency, and the cage harmonics to the fundamental cage frequency. The top three peaks within each of the ranges are identified as possible matches to their respective fundamental frequencies. Out of the three selected peaks, BFE identifies the desired harmonic as the peak whose frequency is the closest multiple to the respective fundamental frequency, given a range of error of $\pm 15\text{Hz}$. In the case where there is not a match between the top three peaks of a harmonic's range with the respective first harmonic frequency, meaning that the given harmonic is not significantly showing in the sensor data, BFE uses the calculated harmonic frequency position and extracts the information for that given harmonic at the calculated frequency.

The reasoning behind looking for the closest multiple instead of an exact multiple is due to the sampling of the high frequency data input to the system, which is set at 5 Hz. A point is picked from each frequency bin as the representative amplitude for the given bin, and the frequency value selected is



A. Sensor Data at 65 Percent Thrust Level.



B. Sensor Data at 109 Percent Thrust Level

Figure 4. Frequency Shift at Different Thrust Level

the halfway point of the bin, making the exact location of a frequency unknown to BFE. The range of $\pm 15\text{Hz}$ allows for a harmonic to be found within three bins to the right or left of the expected harmonic frequency value, due to the error introduced by the sampling rate of the sensor data.

Since the inner and outer race frequencies are exactly determined by a window's fundamental synchronous frequency, the cage to synchronous ratio, and the number of balls in a bearing, they do not go through the same identification process as the synchronous and cage frequencies. To find the inner race frequency information, BFE applies the following

formula:

$$\text{INNER RACE FREQUENCY} = 1X * NB$$

The outer race frequency is computed by almost the same formula, except that instead of multiplying the fundamental cage frequency, BFE uses the complement of that value, or one minus the fundamental cage frequency:

$$\text{OUTER RACE FREQUENCY} = 1N * (1 - (1X/1N)) * NB$$

where

1N = Fundamental Synchronous Frequency

1X = Fundamental Cage Frequency

NB = Number of Balls in a Bearing

Once the frequencies are identified, BFE uses the specific frequency point and extracts the corresponding amplitude value.

The Design of the Intermittent Frequency Extractor (IFE)

Intermittent frequencies are defined as the frequency peaks found in a sensor's data window that cannot be categorized as basic frequencies and are significantly appearing in the data. If any of the frequencies in the data window are selected as intermittent frequencies, they are sent to the Intermittent Frequency Classifier (IFC), either to be identified as one of the possible feed-through frequency signals from the other SSME components, shown in Table 2, or to be classified as unknowns.

KNOWN INTERMITTENT FREQUENCIES
1. 60 Hz LINE
2. HPOTP CASE MODE
3. HPOTP FIRST ROTOR MODE
4. LPOTP SYNCHRONOUS AND HARMONICS
5. LPFTP SYNCHRONOUS AND HARMONICS
6. HPFTP SYNCHRONOUS AND HARMONICS

Table 2. Possible SSME Component Feed-through Frequencies

In determining the intermittent frequencies, IFE employs a threshold significantly above the white noise level. The White Noise Level Extractor (WNLE) determines the white noise level by analyzing a sensor's data window. In order to avoid the monitored frequencies' peaks from influencing the white noise value, they are removed from the data window before the computation; in this way, only the noisy amplitudes contribute to the white noise value. WNLE then averages the remaining amplitude values to determine the white noise level value for a given sensor at each time slice. Because this value is also used as a threshold basis for the extraction of some frequencies, its standard deviation is also computed.

Any frequency whose amplitude is significantly above the white noise level of a sensor's data window is extracted by the Intermittent Frequency Extractor (IFE). In order to distinguish the intermittent frequencies from the noise found in a sensor's current data window, a threshold based on the sensor's white noise level is applied to the remaining frequencies. This threshold is set at three standard deviations above a sensor's current white noise level, and if any frequency's amplitude is above this threshold, it is extracted from the window and sent to the Intermittent Frequency Classifier for possible identification.

The choice for using three standard deviations above the white noise was determined after applying 2 and 3 standard deviations above the noise to the sensor data, and selecting the threshold which best eliminated the noisy frequencies from the ones that are clearly above the noise level, as shown in Figure 5, over a set of data windows.

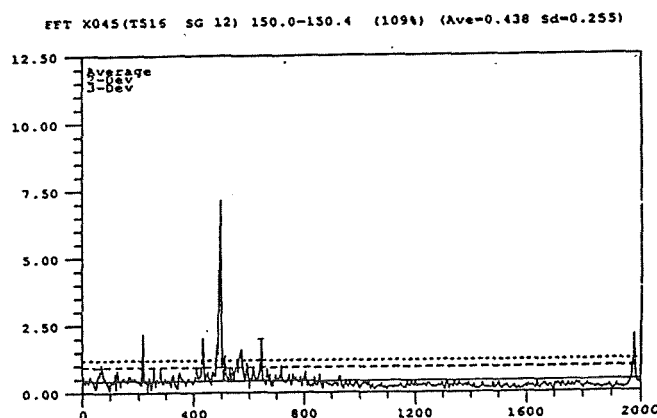


Figure 5. Intermittent Frequency Thresholding Criteria

The Intermittent Frequency Classifier, IFC, attempts to identify the selected frequencies as one of the possible feed-through frequencies that, at times, may appear in the sensor data. Since these frequencies also shift based on SSME's current thrust level, average ratios between the nominal frequencies and the first synchronous frequency were derived from the values used by experts and shown in Table 3, in order to determine the approximate location of the frequency in a data window. These approximate locations are then used by IFC to attempt to match an intermittent frequency as one of the other possible feed-through frequencies.

FREQUENCY NAME	100%	104%	109%
1. LPOTP	84	86	88
2. LPFTP	260	264	273
3. HPOTP	444	470	490
4. HPFTP	572	586	605

Table 3. Feed-Through Frequency Positions at Different Thrusts

The calculated ratios for identifying intermittent frequencies are encoded in a set of data-driven production rules. IFC applies each production rule, computing the approximate ranges for the feed-through frequencies, and attempts to match an incoming intermittent frequency as one of the possible other nominal frequencies. If the incoming frequency does not fit in any of the ranges, then it is classified as an unknown frequency; otherwise, it is identified with the name of the nominal frequency at that range.

Another function IFC performs is to learn intermittent frequencies that it has not seen, and to recognize the ones that it has seen in a previous data window for each of the monitored sensors. From the beginning of a run, IFC starts learning the intermittent frequencies for each of the sensors, and after the first time slice, it starts to recognize previously seen intermittent frequencies. For each intermittent frequency that IFC extracts and recognizes as a previously seen frequency, IFC flags the output information for that frequency as a recognized frequency; otherwise, IFC learns the new frequency for that given sensor so that it can remember it later if the frequency reappears in a different time slice.

V. Training Methodologies

During the development of IDES, several training methods were studied in order to analyze and detect anomalies through sensor data. This section presents and explains each of the approaches that have been tested.

Off-line Training Method

This method involved training the system to distinguish between normal and abnormal sensor data off-line. This was done by using test data which NASA engineers considered normal, and then using that normal data as a basis of comparison with the data from any of the other tests. When comparing the incoming data with the expected normal data in order to detect anomalous frequencies, IDES calculates the number of how many deviations the new frequency amplitude is above the expected normal amplitude.

In order to train the system, the amplitudes for each of the frequencies were extracted from several windows at the different thrust levels present in test 901-039. The mean and standard deviation values were calculated for each of the frequencies at each thrust level. Then, for each of the frequencies, an interpolating function was generated utilizing the different thrust levels and the mean amplitudes at each specific thrust level. In this way, an interpolated mean amplitude can be found for any thrust level, even the ones that were not represented in test 901-039.

Since the abnormality score is based on how many deviations an amplitude value is above the expected interpolated mean, the deviation selected for each frequency was the highest deviation value of that frequency over all of the represented thrust levels. Choosing the highest deviation value for a frequency eliminated the possibility of a normal frequency amplitude being classified as an anomaly in case the selected deviation had been lower than that frequency's deviation at the given thrust level.

In this method the abnormality score was computed by subtracting the interpolated normal amplitude of a frequency from the newly extracted amplitude, and then dividing the result by the highest deviation of that frequency. Anomalous frequencies were detected whenever their deviation number was above a threshold of 5 deviations. Since sensor data have different data scales in different tests (as shown in Figure 6), a scaling factor for each frequency was computed based on the first window of data. In this way, subsequent extracted frequencies' amplitudes were multiplied by their respective scaling factors in order for them to be scaled into representative amplitude values that were comparable to the data scaling of the trained normal amplitudes.

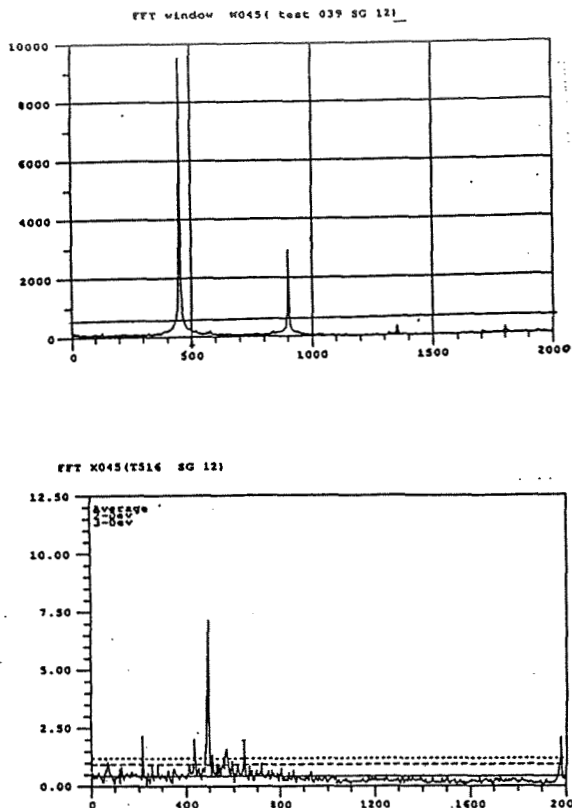


Figure 6. Same Sensor Data From Different Tests.

On-line Training Method

In this approach IDES learned online (from the same test) the expected normal value for each of the frequencies' amplitudes, thus eliminating the need for computing scaling factors between sensor data from different tests. Since the sensor data stabilized within a few seconds after thrust change and can be assumed to be representative of the normal frequency amplitudes, the expected normal values were based on the first window of data, after the sensor data stabilized from a thrust change. Subsequent frequency amplitudes then compared to the amplitudes found in the first data window, and the abnormality scores were determined based on the percentage by which the new amplitude increased from the learned expected normal amplitude. Anomalies were detected whenever a frequency's amplitude exceeded the specified percentage threshold for that specific frequency.

Two improvements were made to the online training algorithm. In the first improvement, the system was modified to compute a running average of the normal amplitude values over a series of data windows. Before adding an amplitude to the running average of a frequency, IDES checked for the

presence of a violation. In the absence of a violation the amplitude was added to the running average; otherwise, the running average was not updated for that time slice. The training terminated when either twenty-five windows had been averaged, or when one of the sensors had detected anomalies in three consecutive windows. In the second modification, the algorithm was changed for the inclusion of non-violating amplitudes in the running average. All the frequencies which did not show anomalous behavior during the training time period, were considered normal and were allowed to update their respective running averages.

Since, in this approach, IDES learned online about the expected normal amplitudes for each of the frequencies, it saved the learned information along with the respective thrust level for possible future use during the test. After each thrust level change, IDES readjusted its expected normal amplitude values to reflect the new thrust level. For every previously unobserved thrust, IDES retrained itself for the data of this new thrust. But if the new thrust level had been observed earlier then the system retrieved the trained amplitude information for that thrust level, rather than learning it again.

Modified On-line Training Method

This approach borrowed concepts from the two previous training and detection methods. It utilized a modified detection criteria for the training algorithm to track all of the frequencies' amplitudes that were being included in each frequency's running average. During the training period, the detection of anomalies was still based on the percentage method, but after the system stops training, the detection of anomalous frequencies was switched to a different algorithm. The mean and standard deviation for each of the frequency's amplitudes were computed during the training period, and the running average values were replaced by the sum of the mean and a multiple of the standard deviation of each frequency after the training period. Detecting anomalies in the data then was based on the new frequencies' amplitudes exceeding the threshold set by the sum of each mean with its respective multiple of the standard deviation.

VI. Sensor Integration Module

When looking at sensor data in an attempt to fuse the information provided from several sensors, experts have always emphasized the consistency of a sensor in detecting an anomalous signal as being very important. The sensor integration algorithm developed for this system uses this consistency heuristic as the basis for generating and integrating the sensors' hypotheses.

The Sensor Integration Module is divided into two parts:

1. Single Sensor Integrator
2. Multiple Sensor Integrator

The first module generates a consistency hypothesis about HPOTP's behavior as seen by a single sensor and based on its monitored frequencies. The second module integrates all the hypotheses generated for all of the sensors into a single overall hypothesis about HPOTP's behavior.

Single Sensor Integrator

The Single Sensor Integrator (SSI) is designed to look at all of a sensor's currently extracted information to see if anomalies have been detected, so that it can combine the new results with the previous ones in order to generate a sensor's consistency hypothesis about possible anomalous behavior of HPOTP. The generated hypothesis is based on a sensor's consistency in detecting anomalous behavior within the same monitored sensor information over a period of three sequential windows.

Each sensor is associated with a symptom object that tracks the consistency of anomalous behavior found in the extracted sensor information along with the hypothesis generated for that given time slice. For each extracted sensor information, there is a sliding window that holds knowledge about whether a sensor has detected an anomaly within the last three time periods. At each new time slice, when a sensor data has been processed and the abnormality scores have been generated, each sliding window is updated to either contain a Y, indicating that an anomaly has been detected at that time for that given sensor information, or an N, indicating that no anomaly has been identified for that given piece of sensor information. In addition, the oldest element found inside each sliding window is removed from it.

Once all of the sliding windows of a sensor's symptom object have been updated, SSI inspects them, giving each sensor a consistency score of either three, two, or zero. As shown in Table 4, at 100.6 seconds a sensor gets a consistency score of three when one of its monitored frequencies has consistently shown an anomaly during three consecutive time slices; in this case, the fundamental synchronous frequency generated the score of three. If a sensor does not receive a consistency score of three, then a consistency score of two is given if one of the frequencies has detected an anomaly in two out of the three tracked time slices, as shown in Table 4 at 98.2 and 98.6 seconds. Otherwise, if a sensor has not received a score of either three or two, then it is assigned a score of zero, indicating that either no anomalies have been detected by the sliding windows at 97.0 seconds in

Table 4, or that there is only one anomaly detected in any of the sliding windows at 97.4 seconds in Table 4.

TIME	1N	2N	3N	4N	1X	2X	3X	4X	1R	OR
96.6										
97.0										
97.4	X									
97.8										
98.2	X									
98.6	X									
99.0										
99.4										
99.8	X									
100.2	X									
100.6	X									

Table 4. A Sensor's Data Behavior.

The Multiple Sensor Integrator

Once all of the sensors have had their consistency hypothesis generated for a given time slice, the symptom objects are sent to the Multiple Sensor Integrator (MSI) to have the overall hypothesis about HPOTP's health status generated for the current time. MSI generates three possible hypothesis:

1. Possibility of a fault
2. Good possibility of a fault
3. Fault occurring based on N consistent sensors

In order for MSI to generate a hypothesis, it must first look into the consistency score found in the symptom objects of each sensor, and then determine how many sensors have received consistency scores of either three or two. For MSI to assign the first hypothesis (the possibility of a fault) as HPOTP's status, only one of the monitored sensors must have received a consistency score of three, indicating that it has consistently detected anomalous HPOTP behavior within three consecutive time slices. For the second hypothesis (a good possibility of a fault) to be assigned, one of the monitored sensors must have received a consistency score of three, while at least one

other sensor has received a consistency score of two, showing that one sensor has consistently detected anomalous behavior in three consecutive time slices, and at least one other sensor must have detected anomalous behavior in two out of three consecutive time periods. When two or more monitored sensors have received a consistency score of three, indicating that at least two sensors have detected anomalous behavior, the third hypothesis (fault occurring based on N consistent sensors) is assigned as HPOTP's status, where N is the number of consistent sensors showing the fault. No hypotheses are generated in the cases where all the sensors have consistency scores of either two or zero.

Examples of the sensor data conditions in which the three hypotheses are generated can be seen in Table 5. At 107.4 seconds, MSI generates the first hypothesis of a possibility of a fault. In this case, the Weld Strain Gauge 3A is the cause for the hypothesis based on its anomalous second harmonic of the synchronous frequency. An example of when the second hypothesis is generated is at 108.2 seconds, where now the Isolator Strain Gauge 12 has detected anomalous behavior in two out of the three time slices. At 108.6, MSI generates the third hypothesis which says that a fault is occurring based on two consistent sensors: Isolator Strain Gauge 12 and Weld Strain Gauge 3A.

	ISO SG 12						ISO SG 90						WELD SG 1A						WELD SG 3A					
TIME	1n	2n	3n	4n	1x	2x	3x	4x	1n	2n	3n	4n	1x	2x	3x	4x	1n	2n	3n	4n	1x	2x	3x	4x
106.6					X				X		X						X							
107.0																								
107.4																	X							
107.8					X						X													
108.2					X												X							
108.6					X	X							X	X			X				X	X		
109.0					X	X							X	X			X				X	X		
109.4					X	X							X	X			X				X	X		

Table 5. Sensors' Time Slice Outputs.

VII. Conclusion

We have developed a prototype of the Identification and Detection Expert System (IDES) which has been tested on the high frequency data of two SSME tests, namely Test \$901-0516 and Test # 904-044. The comparison of our results with the post-test analysis, performed by human experts, indicates the high potential capabilities of IDES in detecting anomalies significantly earlier than other methods currently being applied. However, the implementation of the prototype is the first step in achieving our goals. The success of IDES must be tested on a number of tests

of different faults as well as on the same fault occurring with different severities and speeds. We expect that several modifications will be needed for the successful testing of IDES on the data of a large number of engine tests. Though we have performed preliminary analysis of intermittent frequencies, a large amount of domain knowledge will be needed in order to successfully interpret and employ the analysis of these frequencies to the data of a large number of tests. Furthermore, the identification of patterns and the detection of anomalies in early stage may not be enough. The diagnosis of faults, finding causes and sources of the problem, and determining of the possible corrective actions will be important extensions which we intend to perform in the future.

Acknowledgement

This research was supported by the System Dynamics Laboratory of NASA Marshall Space Flight Center, Grant No. NAG8-121. We thank Pat Valley, Tom Fox, and Carlyle Smith for their assistance throughout this project and Kewei Wu for assisting in data storage and retrieval process.

References

1. Abu-Hakima, S., Davidson, P., Halasz, M. and Phan, S., "Jet Engine Technical Advisor (JETA)," Proceedings of the Second International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Tullahoma, TN, pp. 154-159, June 6-9, 1989.
2. Dietz, W. E., Kiech, E. L. and Ali, M., "Classification of Data Patterns using Autoassociative Neural Network Topology," Proceedings of the Second International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Tullahoma, TN, pp. 1028-1036, June 6-9, 1989.
3. Dietz, W. E., Kiech, E. L. and Ali, M., "Jet and Rocket Engine Fault Diagnosis in Real Time," Journal of Neural Network Computing, vol. 1, no. 1, pp. 5-18, Summer 1989.
4. Dietz, W. E., Kiech, E. L. and Ali, M., "Pattern-Based Fault Diagnosis using Neural Networks," Proceedings of the First International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Tullahoma, TN, pp. 13-23, June 1-3, 1988.

5. Dietz, W. E. and Ali, M., "Qualitative and Temporal Reasoning in Engine Behavior Analysis and Fault Diagnosis," The Third Conference on Artificial Intelligence for Space Applications, Huntsville, AL, November 2-3, 1987.
6. Forsyth, G. F. and Larking, M. D., "Concept Demonstration of the Use of Interactive Fault Diagnosis and Isolation for TF30 Engines," Proceedings of the Second International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Tullahoma, TN, pp. 15-20, June 6-9, 1989.
7. Gupta, U. K. and Ali, M., "LEADER-An Integrated Engine Behavior and Design Analysis Based Real-Time Fault Diagnostic Expert System for Space Shuttle Main Engine," Proceedings of the Second International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Tullahoma, TN, pp. 135-145, June 6-9, 1989.
8. Handelman, D. A. and Stengil, R. F., "Combining Expert System and Analytical Redundancy Concepts for Fault-Tolerant Flight Control," Journal of Guidance, vol. 12, no. 1, pp. 39-45, January-February 1989.